

### A sequential Circuit

A sequential circuit is a logical circuit in which the output depends on the present value of the input signal as well as the sequence of past output.

A sequential circuit is a logical circuit which consists of a combinational circuit to which memory elements are connected to form a feedback path. The memory element are devices capable of storing binary information at any given time.

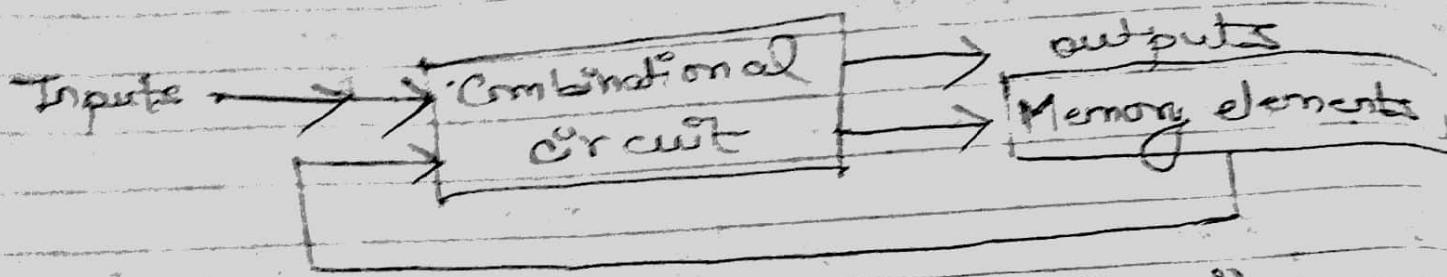


Fig: Block diagram of sequential circuit.

The block diagram of a sequential circuit is shown in figure. The sequential circuit receives binary information from external inputs. These inputs together with the present state of the memory element determine the binary value at the output terminal.

There are two types of sequential circuit.

- Synchronous sequential circuit.

- Asynchronous sequential circuit.

The synchronous sequential circuit is a system whose behavior can be defined from the knowledge of its signal at discrete interval of time. Example flip-flop, synchronous counters.

The Asynchronous sequential circuit is a circuit whose behavior can be defined on the order in which its input signal change and can be affected at any instant of time. Example Asynchronous counter.

- Q. Difference between synchronous and Asynchronous sequential circuit.

#### Synchronous Sequential Circuit

- It is easy to design
- A clocked flip flop acts as memory element.
- They are slower as clock is involved.
- The states of memory element is affected only at active edge of clock, if input is changed.

#### Asynchronous Sequential Circuit

- It is difficult to design
- An unclocked flip flop or time delay is used as memory element.
- They are comparatively faster as no clock is used here.
- The states of memory element will change any time as soon as input is changed.

## Difference between Combinational Circuit & Sequential Circuit

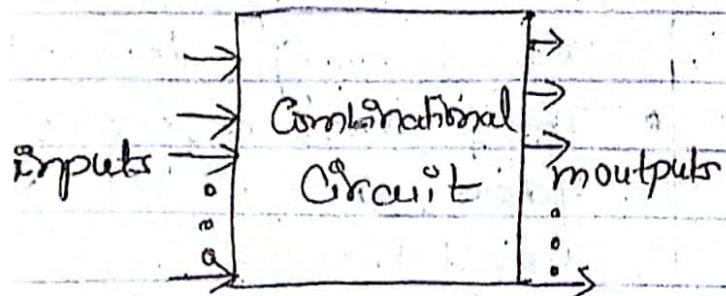
### Combinational Circuits

The circuit whose output at any instant depends only on the input present at that instant only is known as Combinational Circuit.

This type of circuit has no memory unit -

Examples of Combinational Circuits are half adder, full adder, magnitude comparators, multiplexer, demultiplexer, etc.

Faster in speed.



re: Block Diagram of Combinational Circuit

### Sequential Circuit

1. The circuit whose output at any instant depends not only on the input present but also on the past output is known as Sequential Circuit.
2. This type of circuit has memory unit for store past output.
3. Examples of sequential circuits are flip flop, registers, counter, etc.
4. Slower compared to Combinational Circuit.

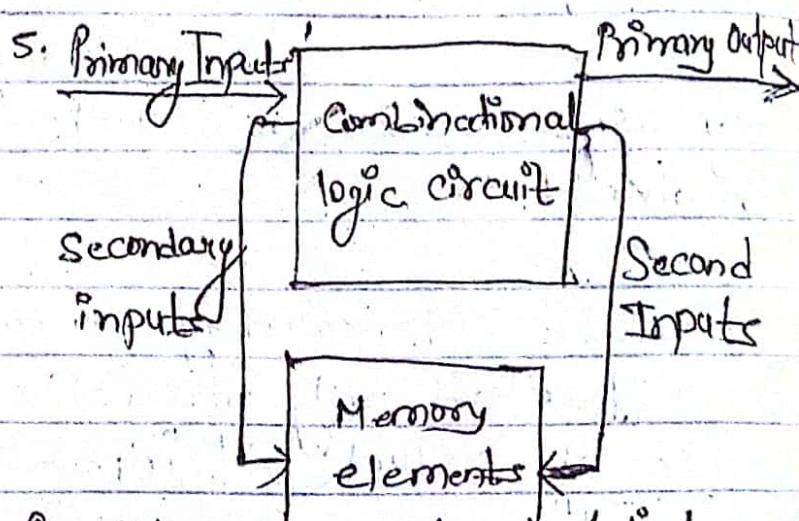
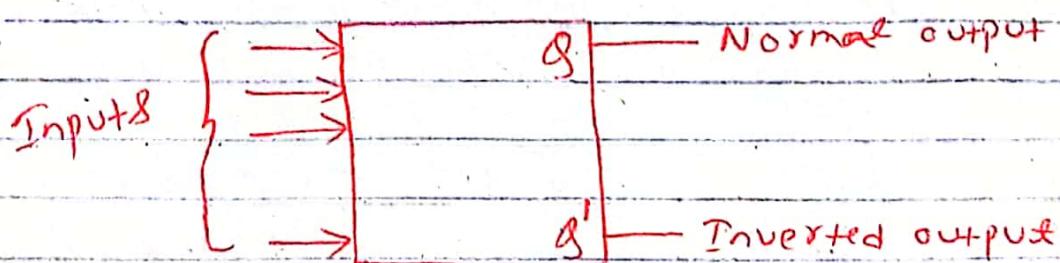


fig: Block Diagram of Sequential Circuit

## The Flip Flop

The memory elements used in clocked sequential circuits are called flip-flops. These circuits are binary cells capable of storing one bit of information. A flip flop circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it. A flip flop is also known as a bistable multivibrator. Flip flop can be obtained by using NAND or NOR gates.

The general block diagram representation of a flip flop is shown in figure(a). It has one or more inputs and two outputs. The two outputs are complementary to each other. If  $Q$  is 1 i.e Set, then  $Q'$  is 0; if  $Q$  is 0 i.e Reset, then  $Q'$  is 1. That means  $Q$  and  $Q'$  cannot be at the same state simultaneously.



fig(a) Block diagram of a flip flop

~~H~~ Sequential logic

## Latch vs Flip flop

### Latch

1) Latches do not require clock signal.

2) A latch is an asynchronous device.

3) Latches are simpler to design as there is no clock signal.

4) The operation of latch is faster as they do not have to wait for any clock signal.

5) The power requirement of a latch is less.

6) A latch works based on the enable signal.

7) Latch continuously checks its inputs and changes its output correspondingly.

### Flip flop

1) Flip flop have clock signal.

2) A flip flop is a synchronous device.

3) Flip flop are more complex design as they have clock signal and it has to be carefully routed.

4) Flip flops are comparatively slower than latches due to clock signal.

5) The power requirement of a flip flop is more.

6) A flip flop works based on the clock signal.

7) Flip flop continuously checks its inputs and changes its output correspondingly only at times determined by clocking signal.

## Types of Flip-Flop

- 1) S-R Flip Flop
- 2) D - Flip Flop
- 3) JK - Flip Flop
- 4) T - Flip Flop

### D) S-R Flip Flop (Set-Reset flip-flop)

An S-R flip flop has two inputs named Set (S) and Reset (R), and two outputs Q and  $Q'$ . The outputs are complement to each other i.e. if one of the output is 0 then the other should be 1. This can be implemented with NAND or NOR gates.

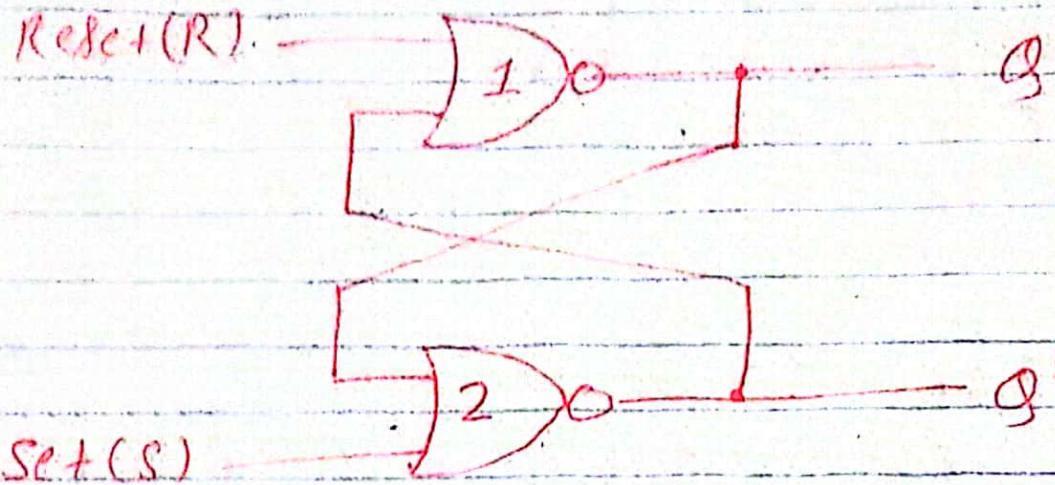
The block diagram of an S-R flip flop is given as:-



fig(a): S-R flip flop

#### a) S-R Flip Flop Using NOR gates

An S-R flip flop can be constructed with NOR gates at ease by connecting the NOR gates back to back as shown in fig (b).



fig(3): S-R FLIP using NOR gates.

The cross-coupled connections from the output of gate 1 to the input of gate 2 constitute a feedback path. The truth table for the S-R flipflop based on a NOR gate is shown in table (a):

Inputs		outputs		ACTION
S	R	$Q_{n+1}$	$Q'_n+1$	
0	0	$Q_n$	$Q'_n$	No change (memory)
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Forbidden (Undefined)

To analyze the circuit, we have to consider the fact that the output of a NOR gate is 0 if any of the inputs are 1, irrespective of other input. The output is 1 only if all of the inputs are 0.

CASE 1: When  $S=0$  and  $R=0$ , the output is  $Q$  and  $Q'$  i.e. the flip flop remains in its present state. It means that the next state of the flip flop does not change, i.e.  $Q_{n+1} = 0$  if  $Q_n = 0$  and vice versa. This is the memory condition of the S-R flip flop.

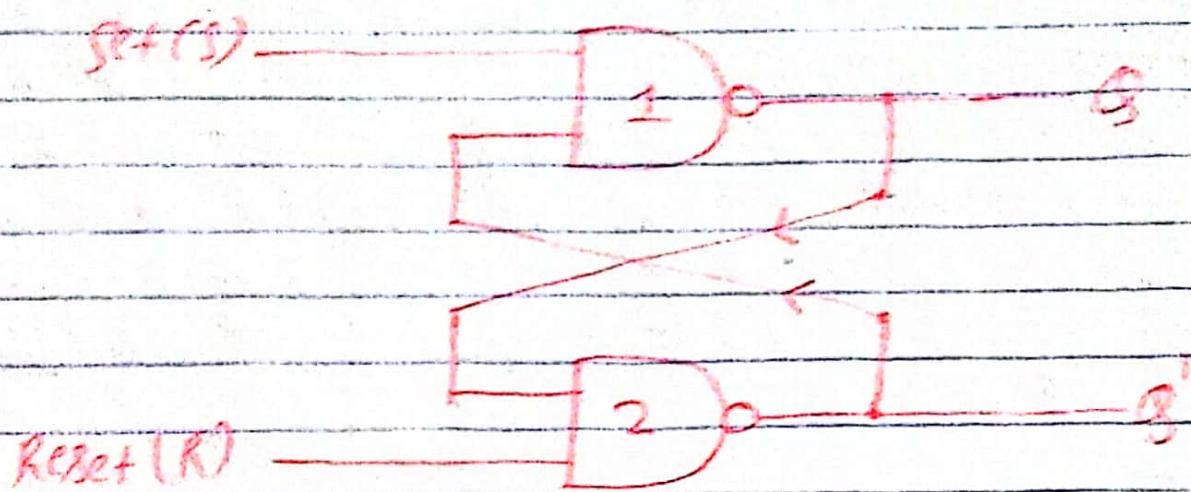
CASE 2: When  $S=0$  and  $R=1$ , the output becomes  $Q=0$  and  $Q'=1$ . The 1 at R input forces the output of NOR gate 1 to be 0 (i.e.  $Q_{n+1}=0$ ). Hence both the inputs of NOR gate 2 are 0 and 0 and so its output  $Q'_{n+1}=1$ . Thus the condition  $S=0$  and  $R=1$  will always reset the flip flop to 0.

CASE 3: When  $S=1$  and  $R=0$ , the output becomes  $Q=1$  and  $Q'=0$ . The 1 at S input forces the output of NOR gate 2 to be 0 (i.e.  $Q'_{n+1}=0$ ). Hence both the inputs of NOR gate 1 are 0 and 0 and so, its output  $Q_{n+1}=1$ . Thus, the condition  $S=1$  and  $R=0$  will always set the flip flop to 1.

CASE 4: When  $S=1$  and  $R=1$ , the output becomes  $Q=0$  and  $Q'=0$  (invalid). The 1 at S input and R input forces the output of both NOR gate 1 and NOR gate 2 to be 0 i.e.  $Q_{n+1}=0$  and  $Q'_{n+1}=0$ . This condition  $S=1$  and  $R=1$  violates the fact that outputs of a flip flop will always be the complement of each other so, this is invalid state and must be avoided.

### 5) S-R flip flop using NAND gates

An S-R flip flop can be constructed with NAND gates by connecting the NAND gates back to back as shown in fig(c). The operation of the flip flop can be analyzed in a similar manner that employed for the NOR gated S-R flip flop.



fig(c): NAND based S-R flip flop

To analyze the circuit, we have to remember that a low (0) at any input of a NAND gate forces the output to be high (1), irrespective of the other input. The output of a NAND gate is only if all of the inputs of the NAND gate are high (1).

The output truth table for the S-R flip flop based on NAND gate is shown in table (3).

Inputs		Outputs		Action
S	R	$Q_{n+1}$	$Q'_{n+1}$	
0	0	1	1	Forbiting (undefined)
0	1	1	0	Set
1	0	0	1	Reset
1	1	$Q_n$	$Q'_n$	No change

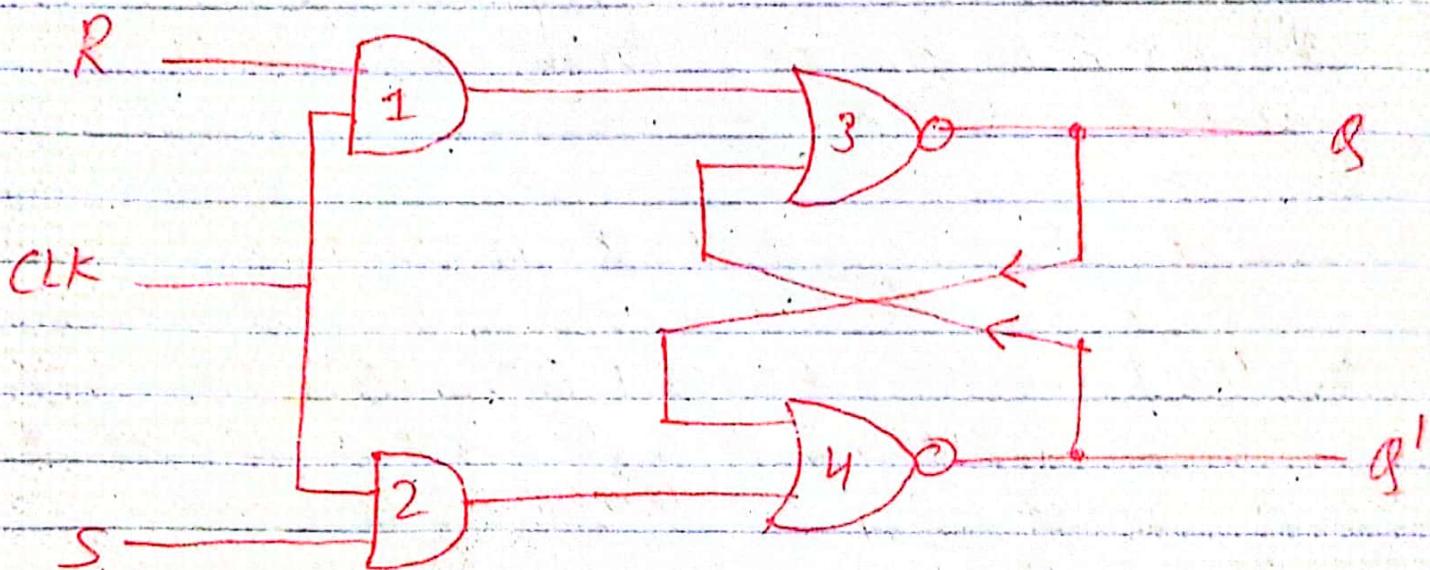
Case 1: When  $S=0$  and  $R=0$ , the output becomes  $Q=1$  and  $Q'=1$  (invalid). The 0 at R input and S input forces the output of both NAND gate 1 and NAND gate 2 to be 1. i.e  $Q_{n+1}=1$  and  $Q'_{n+1}=1$ . This condition  $S=0$  and  $R=0$  violates the fact that the outputs of a flip-flop will always be the complement of each other. Since the condition violates the basic definition of a flip-flop, it is called the undefined condition. Generally, this condition must be avoided by making sure that 0s are not applied simultaneously to both of the inputs.

Case 2: When  $S=0$  and  $R=1$ , the output becomes  $Q=1$  and  $Q'=0$ . The 0 at S input forces the output of NAND gate 1 to be 1. (i.e  $Q_{n+1}=1$ ). Hence both the inputs of NAND gate 1 and 2 are 1 and so its output  $Q'_{n+1}=0$ . Thus, the condition  $S=0$  and  $R=1$  will always set the flip-flop to 1.

Case 3: When  $S=1$  and  $R=0$ , the output becomes  $Q=0$  and  $Q'=1$ . The '0' at R input forces the output of NAND gate 2 to be 1 (i.e.  $Q_{n+1}=1$ ). Hence both the inputs of NAND gate 1 are 1 and its output  $Q_n=0$ . Thus, the condition  $S=1$  and  $R=0$  will always reset the flip-flop to 0.

Case 4: When  $S=1$  and  $R=1$ , the output becomes  $Q_n=1$  and  $Q'=1$  i.e. the flip-flop remains in its present state ( $Q_n$ ). It means that the next state of the flip-flop does not change i.e.  $Q_{n+1}=0$  if  $Q_n=1$  and vice versa. This is the memory condition of the SR flip-flop.

### c) Clocked S-R Flip flop

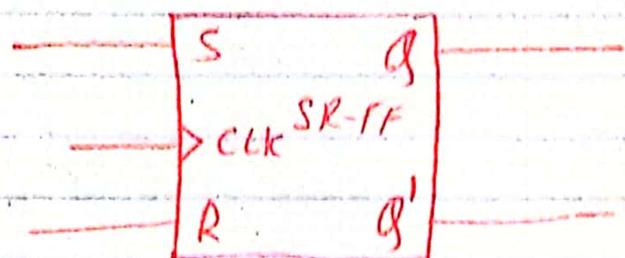


fig(d): Clocked S-R flip flop.

~~Ques~~

The Clocked SR flip flop is shown in the figure (a). It consists of a basic NOT flip flop and two AND gates. The clock input is connected to both of the AND gates. The output of two AND gates remain at '0' as long as the clock pulse (cp/clk) is '0' regardless of the S and R inputs. In this situation, the changes in S and R inputs will not affect the state ( $Q$ ) of the flip flop. On the other hand, if the clock input is high (1), the changes in S and R will be passed over by the AND gates and they will cause changes in the output ( $Q$ ) of the flip flop.

The logic graphic symbol of the SR flip flop is shown below. It has three inputs: S, R and CLK (cp). The CLK input is marked with a small triangle. The triangle is a symbol that denotes the fact that the circuit responds to an edge or transition at CLK input.



The output of the flip flop are marked as  $Q$  and  $Q'$  within the box. The state of the FF

$Q_n$  determined from the value of 1st normal output  $Q^1$ . To obtain the complement of a normal output it is not necessary to insert a Inverter because the complemented value is available directly in  $Q^1$ .

- The Set State is reached with  $S=1$ ,  $R=0$  and  $CP$
- To change to the clear state, the inputs must be  $S=0$ ,  $R=1$  and  $CP=1$ .
- with both  $S=1$  and  $R=1$ , the occurrence of  $CP$  causes both outputs to momentarily go to '0'.
- When the clock pulse is removed, the state of the flipflop is Indeterminant.

Flip flop inputs		Present output	Next output
S	R	<del>Q<sub>n</sub></del> $Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

The Characteristic table of SR flip-flop is shown above.  $Q_n$  is the binary state of the

254

flipflop at a given time (present state). The S and R column gives the possible values of the input and  $Q_{n+1}$  is the state of flipflop after occurrence of a clocked pulse (next state).

From characteristic table, we can find out the characteristic equation with the help of K-map as:-

$S \setminus RQ_n$	$R'Q_n$	$RQ_n$	$RQ'_n$	$Q'_n$
$S'$	0	1	0	0
$S$	1	1	X	X

$$Q_{n+1} = S + R'Q_n$$

The excitation table for S-R flipflop is given as:-

Present State	Next State	S-R FLIPFLOP	
$Q_n/Q_{n+1}$	$Q_{n+1}/Q_{n+1}$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

~~D-FLIPFLOP~~

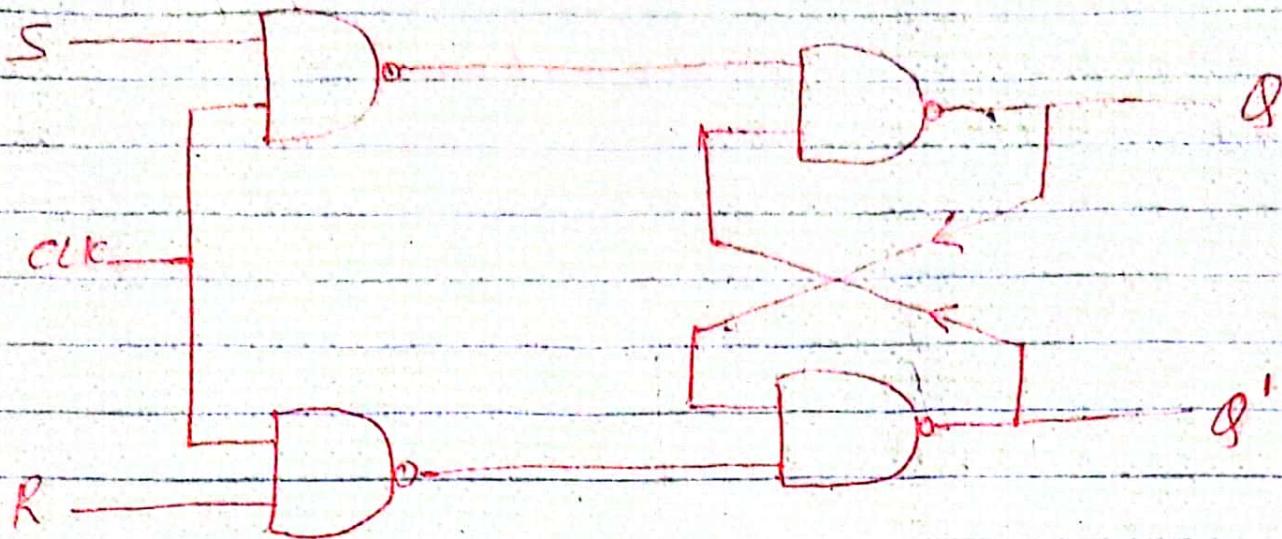


fig (e): Clocked SR flip flop

~~D- FLIPFLOP~~

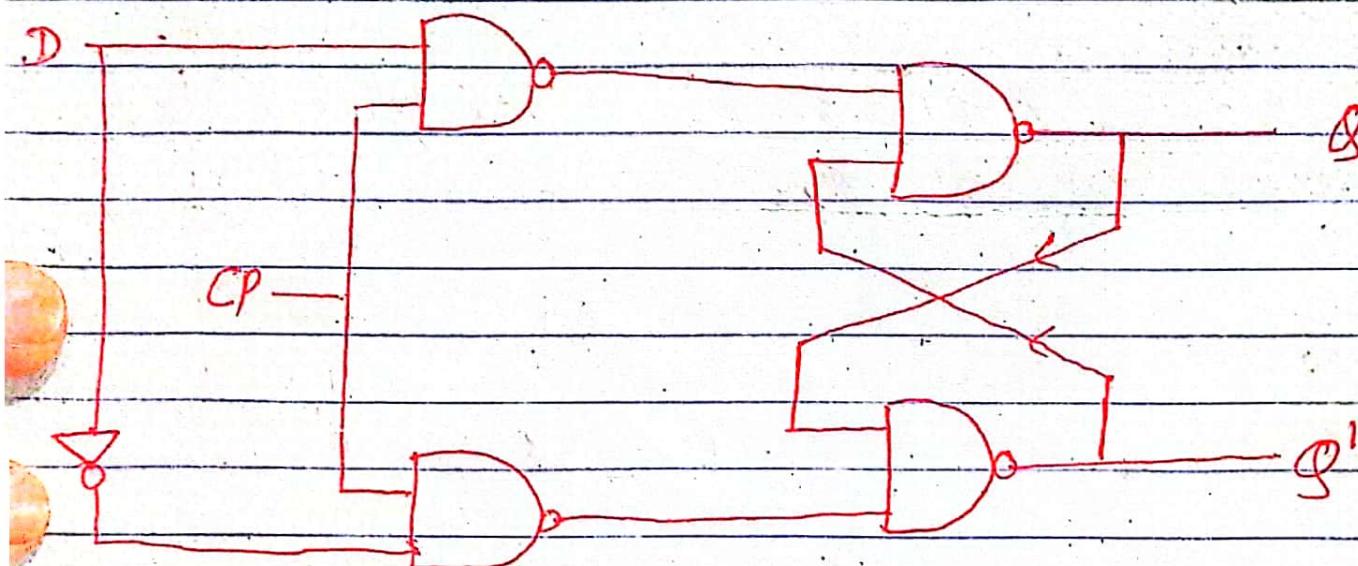


fig (a): Logic diagram of D-flip flop  
using NAND gate.

One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that inputs S and R are never equal to 1 at the same time. This is done in the D latch. The D flip flop has only one input referred to as the D (data) input and two outputs as usual Q and  $\bar{Q}$ . It transfers the data at the input after the delay of one clock pulse at the output Q. So in some cases the input is referred to as a delay input and the flip flop gets the name delay(D) flip flop.

D-flip flop can be easily constructed from an S-R flip flop by simply incorporating an inverter between S and R such that the input of the inverter is at the S end and the output of the inverter is at the R end.

The D-input is passed onto the flip flop when the value of CP is '1'.

When CP is high (1), the flip flop moves to the SET State. If CP is low (0), the flip flop switches to the CLEAR/RESET State.

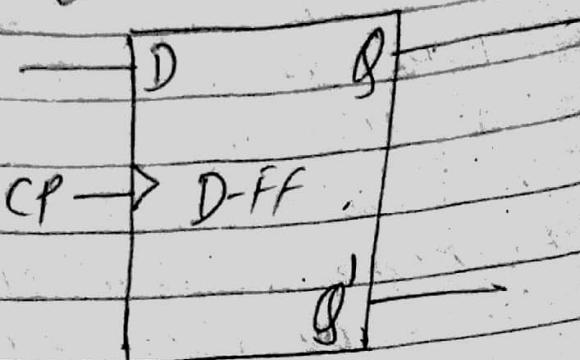
As long as the clock input (CP = 0), the SR latch has both inputs equal to 0 and it can't change its state regardless of the value of D.

When CP is 1, the latch is placed in the set or reset state based on the value of D.

257

If  $D=1$ , the  $Q$  output goes to 1.  
If  $D=0$ , the  $Q$  output goes to 0.

The graphical symbol of a D-flipflop is shown in figure below:



The characteristic table of a D-flipflop is given in the table below:

FF Inputs	present output	Next output
D	$Q_n$	$Q_{n+1}$
0	0	0
0	1	0
1	0	1
1	1	1

From table, we can find out the equation using k-map as:-

D	$Q_n$	$Q'_n$	Q
D'	0	0	
D	1	1	

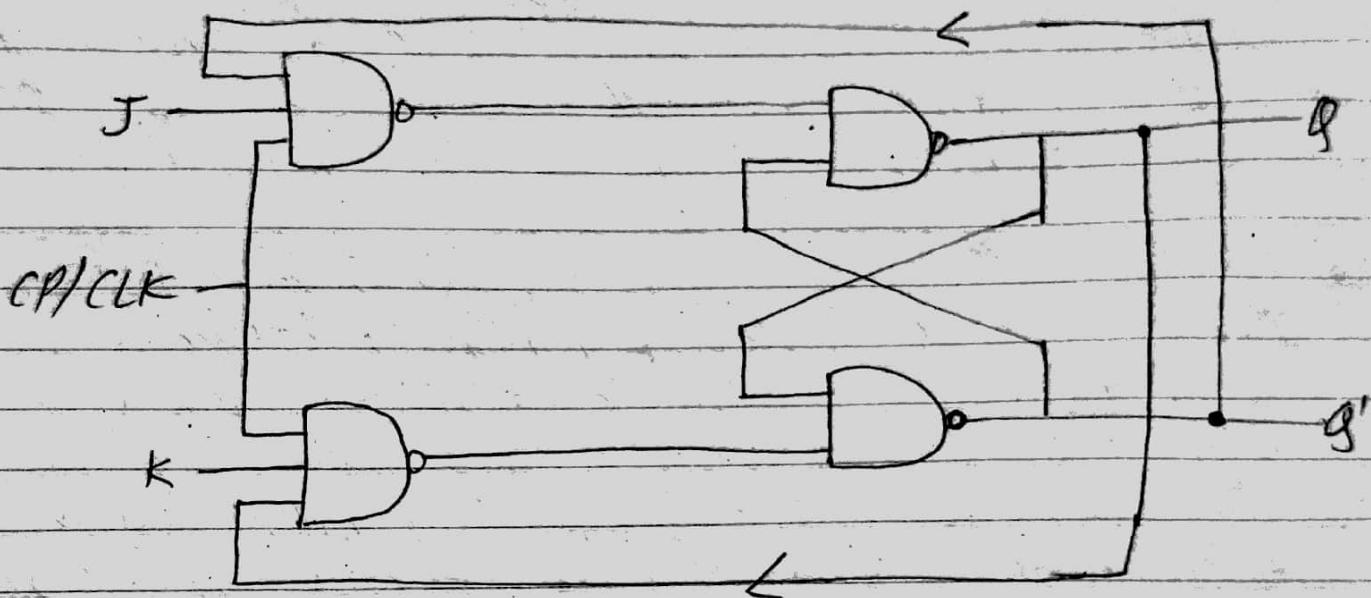
258

$$Q_{n+1} = D$$

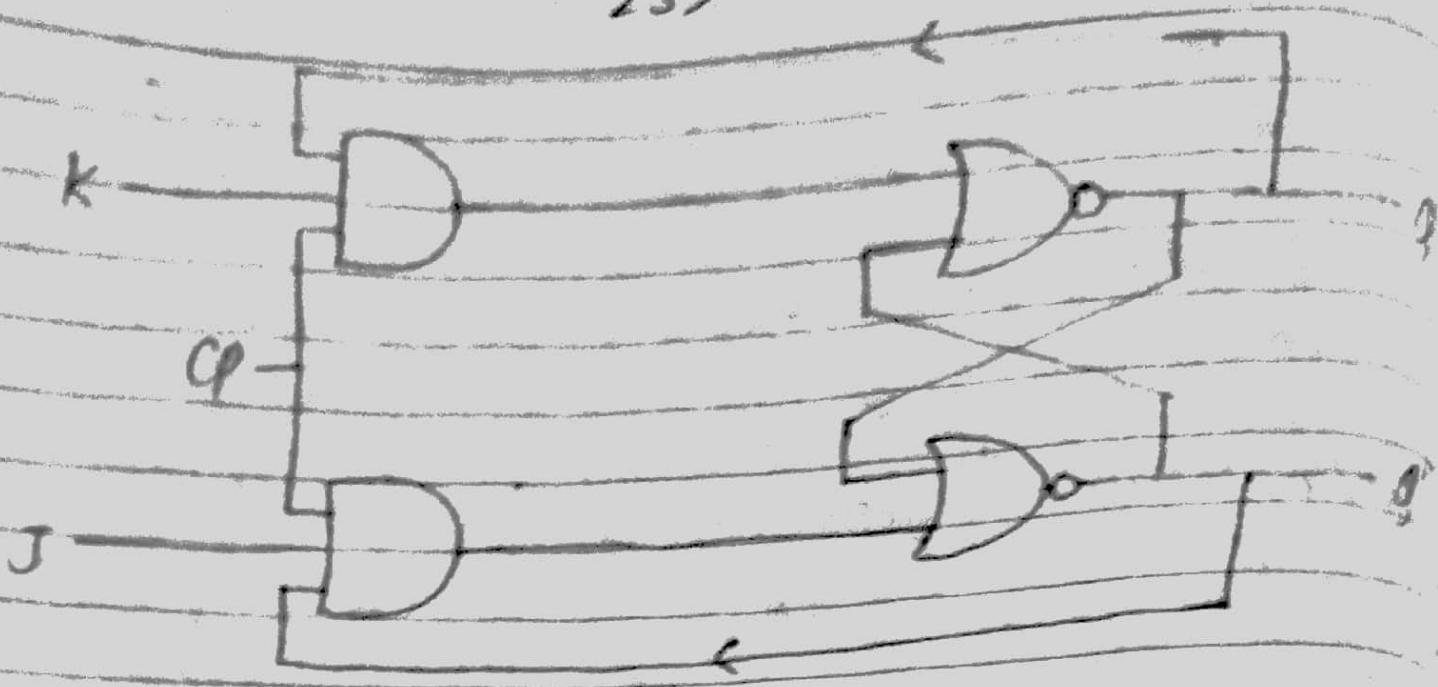
The excitation table for the D-flip flop is given as:-

present state $Q_n$	Next state $Q_{n+1}$	D-FF
0	0	0
0	1	1
1	0	0
1	1	1

### 3) J-K Flip Flop



fig(a) J-K Flip flop using NAND gate.



fig(3) J-K flipflop using NOR and AND gate.

The JK flipflop is an improvement on the SR flipflop.

When both the inputs J and K have a high (1) state, the flip flop switches to the complement state. So, for a value of  $Q=1$ , it switches to  $Q=0$  and for value of  $Q=0$ , it switches to  $Q=1$ .

The circuit includes two 3-input AND gates. The output  $Q$  of the flip flop is returned back as a feedback to the input of the AND along with other inputs like K and clock pulse (CP). Similarly, the output  $Q'$  of the flip flop is given a feedback to the input of the AND along with other inputs like J and clock pulse (CP).

260

If J and K data inputs are different (i.e. high(1) or low(0)) then the output Q takes the value of J at the next clock edge.

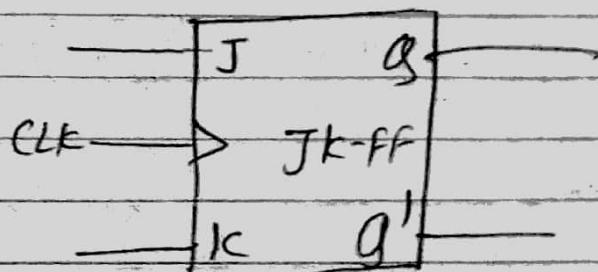
If J and K are both low then no change occurs.

If J and K are both high at the clock edge then the output will toggle from one state to the other.

The truth table for JK flip flop is given as -

Inputs		Output
J	K	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$Q'_n$

The graphical symbol of a JK flip flop is



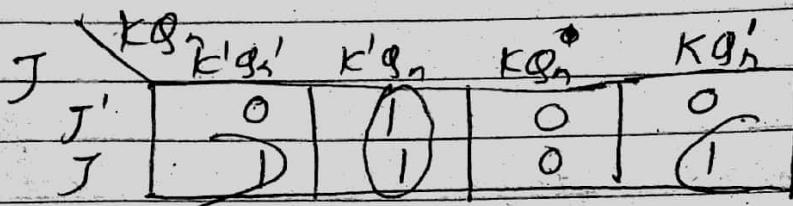
261

The characteristic table of J-K flipflop is given as:

Flip flop inputs		Present output	Next output
J	K	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

4)

The characteristic equation of a J-K flipflop is obtained using K-map as:-



$$Q_{n+1} = \cancel{J'Q_n} + K'Q_n + JQ_n'$$

$$\boxed{Q_{n+1} = JQ_n' + K'Q_n + JKQ_n'}$$

262

The excitation table for the J-K flip flop is shown below:

Present State	Next State	J-K Flipflop	
$Q_n$	$Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

ii) T-Flipflop

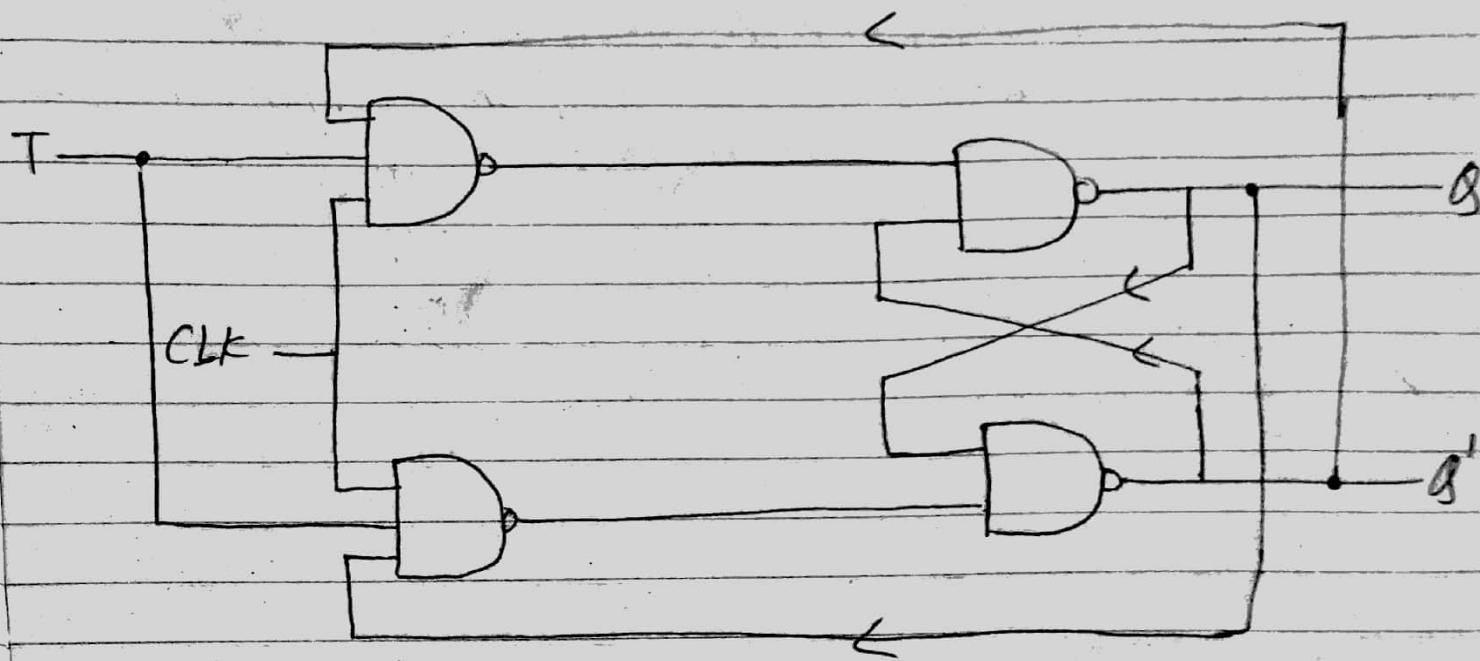


fig: T-Flipflop using NAND gate.

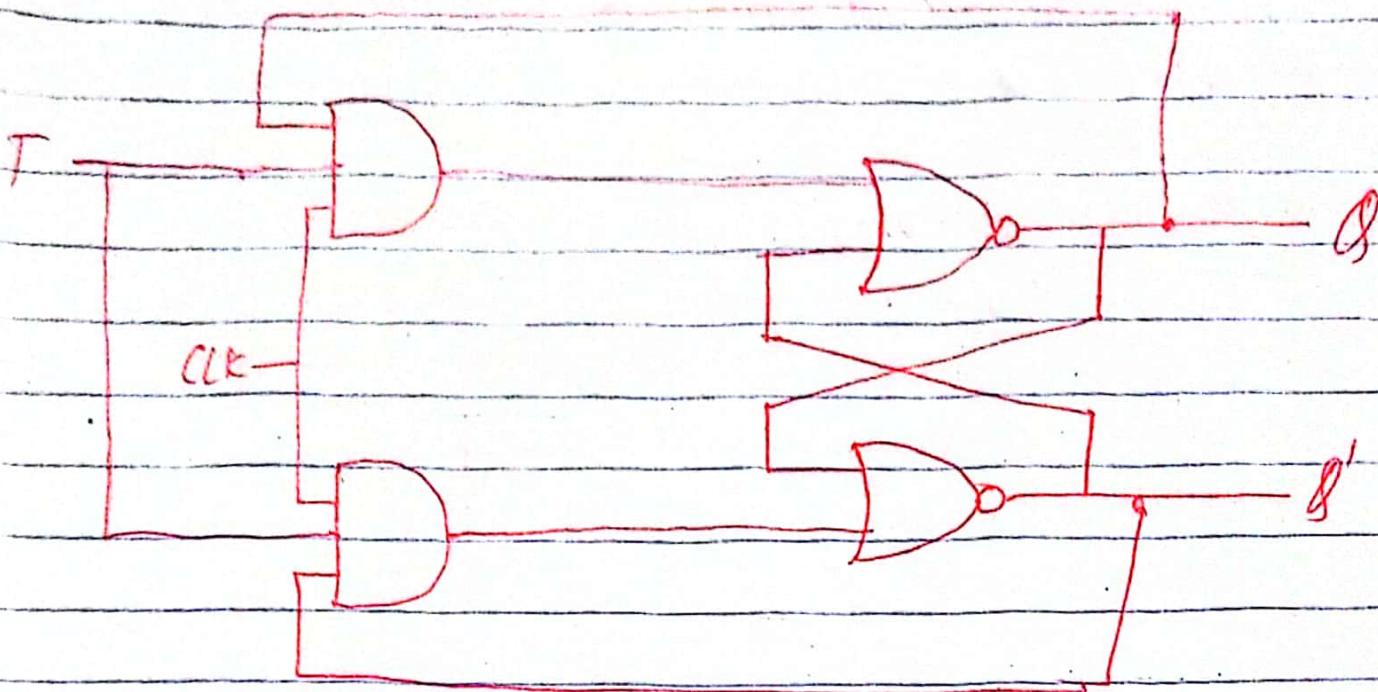


fig: T-Flipflop using NOR gate and AND gate.

The T-Flipflop is a single input version of JK flipflop and is obtained from JK flipflop where both inputs are tied together.

The truth table of a T-Flipflop is given below:-

T	Q <sub>n</sub>	Q <sub>n+1</sub>
0	0	0
0	1	1
1	0	1
1	1	0

The designation 'T' comes from the ability of the flipflop to toggle or change state.

264

Regardless of the present state, the flip flop complements its output when the clock pulse occurs while input T is 1.

The characteristic equation is obtained using k-map

T	$Q_n$	$Q'_n$
T'	0	1
T	1	0

$$\therefore Q_{n+1} = TQ'_n + T'Q_n$$

The characteristic table and equation show that:

- When  $T=0$ ,  ~~$Q_{n+1}=Q_n$~~  i.e. the next state is the same as the present state and no change occurs.
- When  $T=1$ ,  $Q_{n+1}=Q'_n$  i.e. the state of the flip flop is complemented.

The graphical symbol of T-Flipflop is.



# Triggering of Flip flop  
 The state of a flip flop is switched by a momentary change in the input signal. This momentary change is called a trigger, and the transition it causes is said to trigger the flip flop. Clocked flip flops are triggered by pulses. A pulse starts from an initial value of 0, goes momentarily to 1, and after a short time, returns to its initial 0 value.  
 There are many types of triggering. They are...

### i) High Level Triggering

When a flip flop is required to respond at its High State, a High level triggering method is used. It is mainly identified from the straight lead from the clock input. Take a look at the symbolic representation shown below.

Triggers on high level

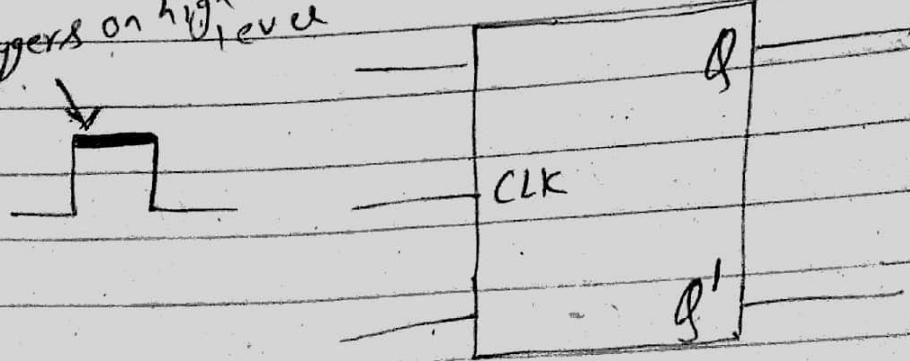


fig: High Level Triggering

### Low Level Triggering

When a flip flop is required to respond at its Low State, a low level triggering method is used. It is mainly identified from the clock input lead along with a low state indicator bubble. Take a look at the symbolic representation shown below:

*triggers on low level*

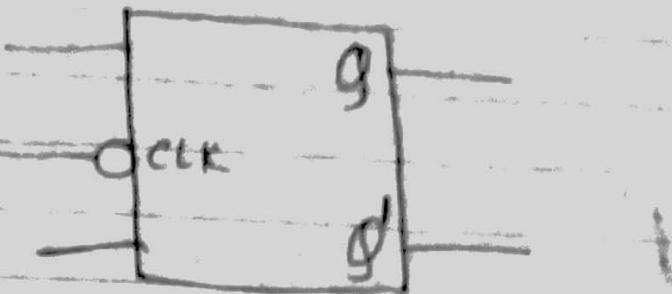


fig: Low Level Triggering

### 3) positive edge triggering

When a flip flop is required to respond at a low (0) to high (1) transition state, positive edge triggering method is used. It is mainly identified from the clock input lead along with a triangle. Take a look at the symbolic representation shown below:

*triggers on this edge of the clock pulse*

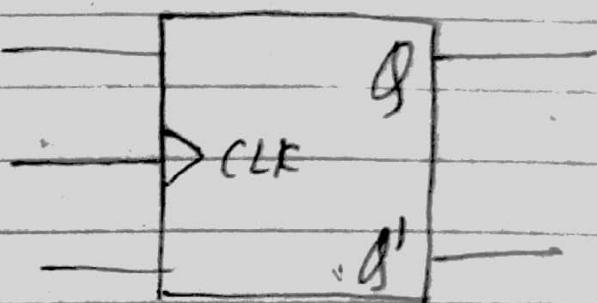


fig: Positive edge triggering

#### 4) Negative Edge Triggering

When a flip flop is required to respond during the High to low transition states a negative edge triggering method is used. It is mainly identified from the clock input lead along with a low-state indicator and a triangle. Take a look at the symbolic representation shown below.

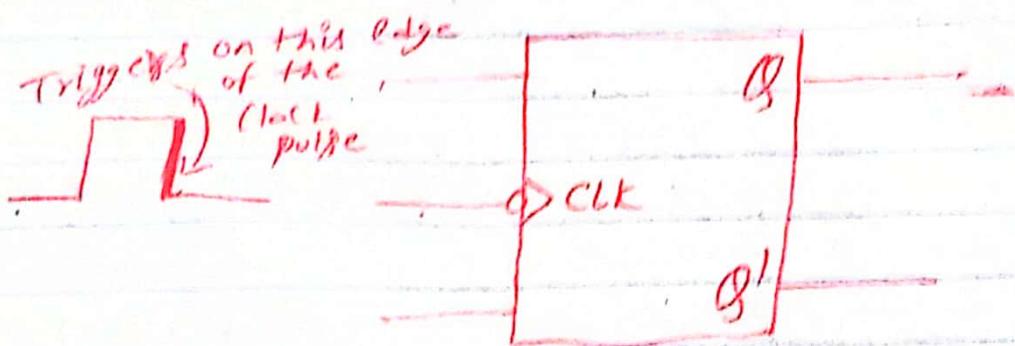


fig: Negative Edge Triggering .

#### # CLOCK PULSE TRANSITION

The movement of a trigger pulse is always from a 0 to 1 and then 1 to 0 of a signal. Thus, it takes two transitions in a single signal. When it moves from 0 to 1, it is called a positive transition and when it moves from 1 to 0, it is called a negative transition.



tve edge  
(-)Negative edge

-ve edge tve edge

~~imp~~ Need for counter design. 2.68

## # FLIPFlop Excitation Table

1)	present state ( $Q_n$ )	Next State ( $Q_{n+1}$ )	S	R
0	0	0	0	X
0		1	1	0
1		0	0	1
1		1	X	0

2)	present state ( $Q_n$ )	Next State ( $Q_{n+1}$ )	J	K
0	0	0	0	X
0		1	1	X
1		0	X	1
1		1	X	0

3)	present state ( $Q_n$ )	Next State ( $Q_{n+1}$ )	D
0	0	0	0
0		1	1
1		0	0
1		1	1

4)	present state ( $Q_n$ )	Next State ( $Q_{n+1}$ )	T
0	0	0	0
0		1	1
1		0	1
1		1	0

## # Master-Slave flipflop

A master-slave flipflop is constructed from two separate flipflops. One circuit serves as a master and the other as a slave, and the overall circuit is referred to as a master-slave flipflop.

### D) RS master-slave flipflop

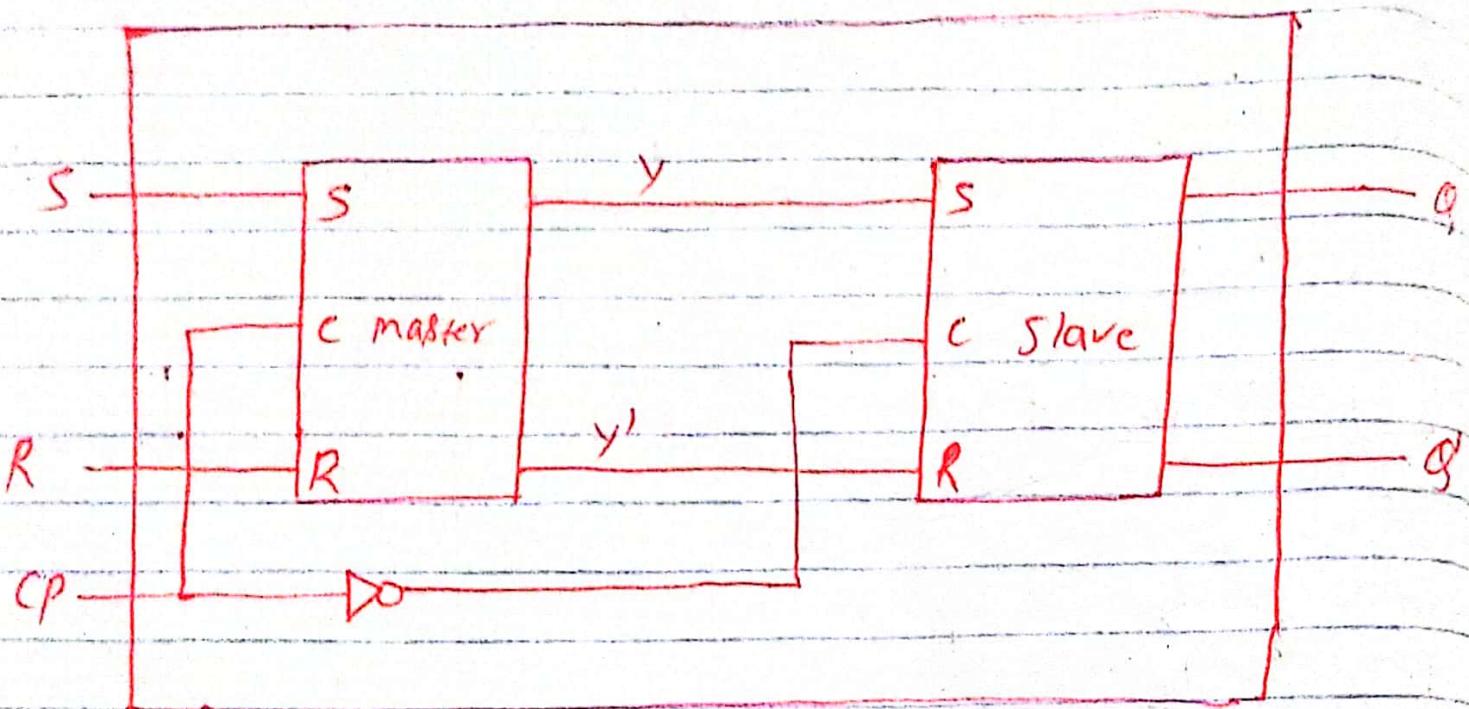


fig: Logic diagram of RS master-slave flipflop

It consists of a master flip flop, a slave flip flop and an inverter. When clock pulse (CP) is 1, the output of the inverter is 1. Since the clock input of the slave is 1, the flip flop is enabled and output Q is equal to Y, while Q' is equal

to 0'. The master flip flop is disabled because CP=0. When the pulse becomes 1, the information then at the external R and S inputs is transmitted to the master flip flop. The Slave flip flop, however, is isolated as long as the pulse is at its 1 level, because the output of the inverter is 0. When the pulse returns to 0, the master flip flop is isolated; this prevents the external inputs from affecting it. The slave flip flop then goes to the same state as the master flip flop.

2) JK Master slave flip flop

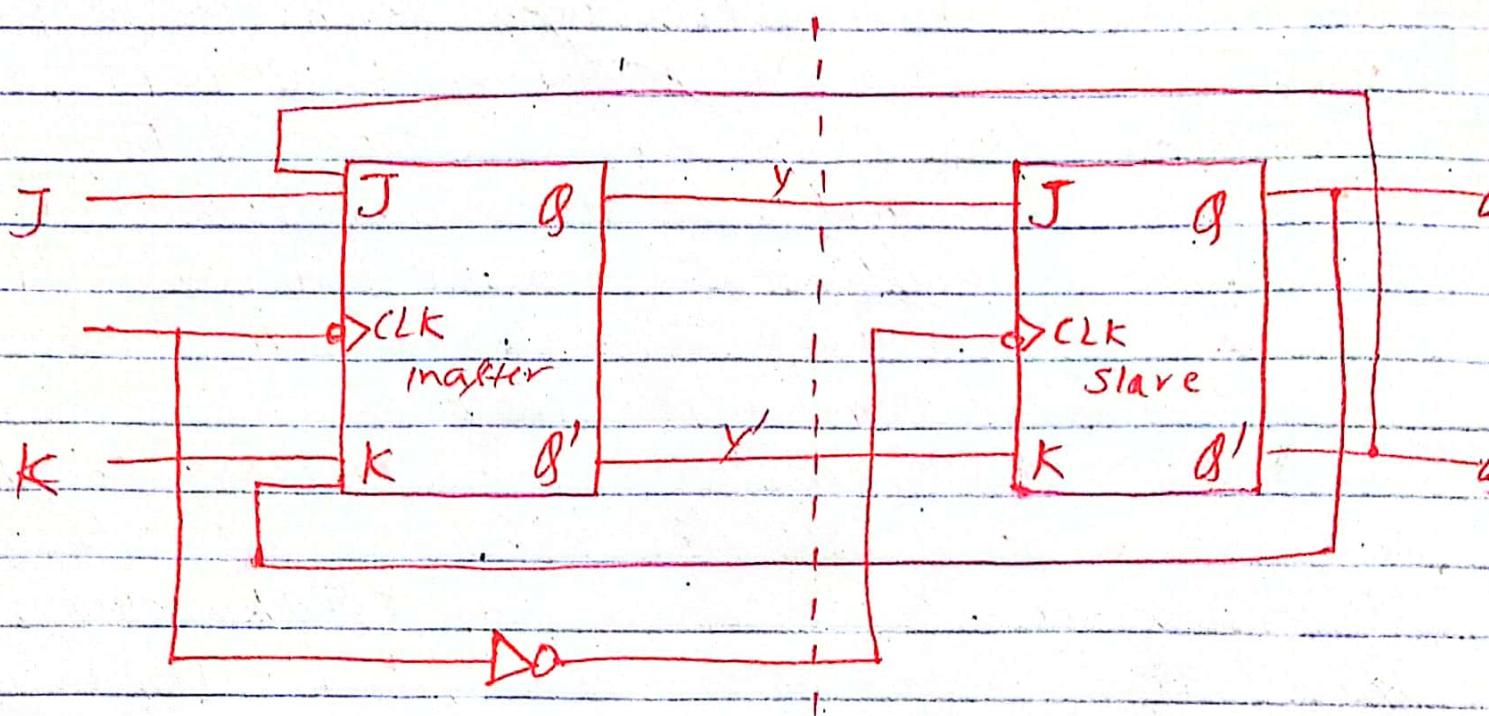


fig: JK Master Slave Flip flop

271

From the figure, we can see that both the J-K flip flops are presented in a series connection. The output of the master J-K flip flop is fed to the input of the slave J-K flip flop. The output of the slave J-K flip flop is given as a feedback to the input of the master J-K flip flop. The clock pulse (CLK) is given to the master J-K flip flop and it is sent through a NOT gate and thus inverted before passing it to the slave J-K flip flop.

### Working

When  $CLK=1$ , the master J-K flip flop gets disabled. The clock input of the master input will be the opposite of the slave input. So, the master flip flop output will be recognized by the slave flip flop only when the clock value becomes 0. Thus, when the clock pulse makes a transition from 1 to 0, the locked outputs of the master flip flop are fed through to the inputs of the slave flip flop making this flip flop edge or pulse triggered.

Thus, the circuit accepts the value in the input when the clock is HIGH and passes the data to the output on the falling edge of the clock signal. This makes the master-slave J-K flip flop a synchronous device as it only passes data with the timing of the clock signal.

## # Application of flip flop

### 1) parallel Data storage

A common requirement in digital system is to store several bits of data from parallel lines simultaneously in a group of flipflops. This operation is illustrated in figure (a) using four flipflops. Each of the four parallel data lines is connected to the D input of a flip flop. The clock inputs of the flipflops are connected together, so that each flipflop is triggered by the same clock pulse. In this example, positive edge triggered flipflops are used, so the data on the D inputs are stored simultaneously by the flipflops on the positive edge of the clock.

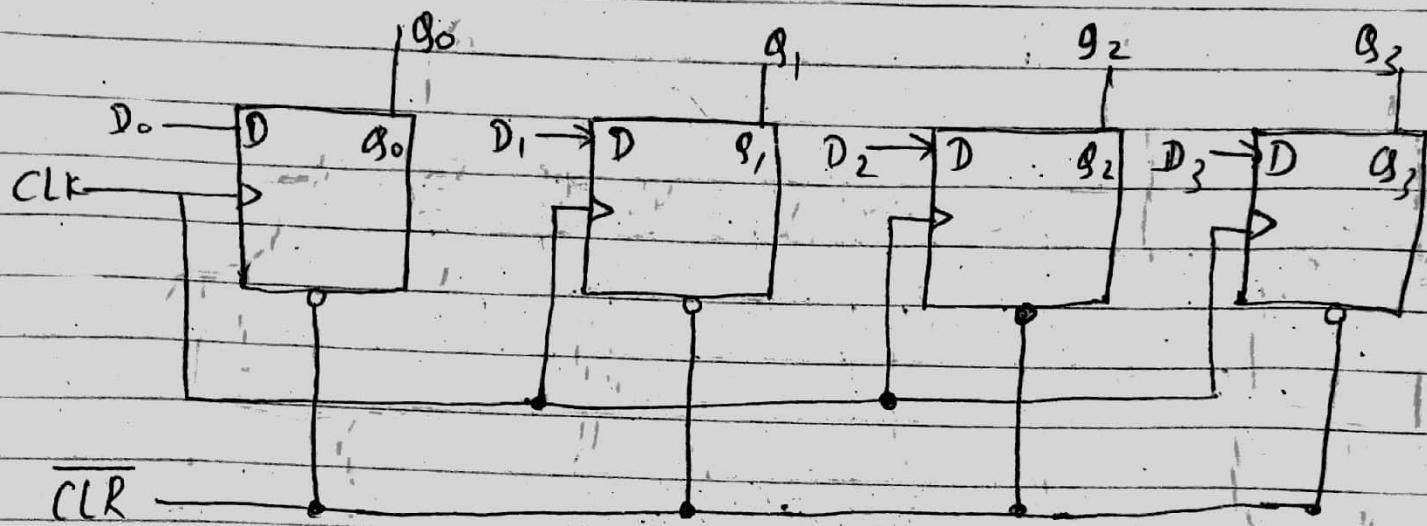


fig: parallel data storage using D-flip-flop

This group of four flip flops is an example of a basic register used for data storage.

## 2) Frequency Division

Another application of a flip flop is division (reducing) the frequency of a periodic waveform. When a pulse waveform is applied to the CLOCK input of a J-K flip flop that is connected to toggle ( $J=K=1$ ), the Q output is a square wave with one half the frequency of the clock input. Thus, a single flip flop can be applied as a divide-by-2 device, as illustrated in figure. As you can see, the flip flop changes state on each triggering clock.

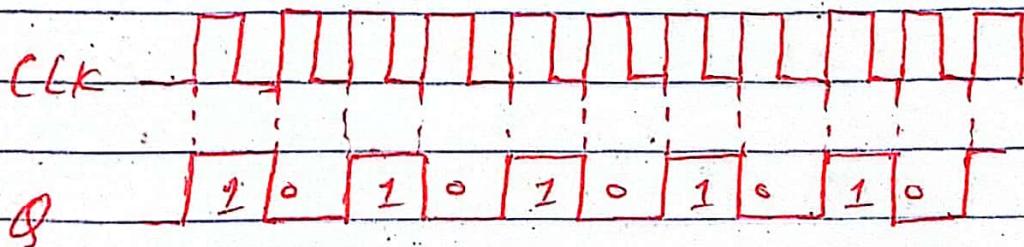
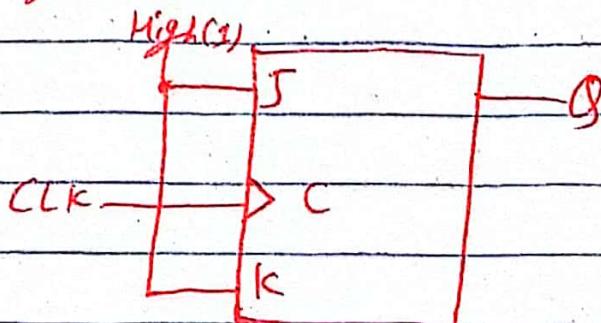


fig: frequency divide by 2 using JK flip flop

## 3) Counting

The flip flops are used for counting sequence

## 4) Shift registers

The flip flops are used to design different shift registers i.e.  $\text{FSO}$ ,  $\text{SISO}$ ,  $\text{PISO}$  and  $\text{PIO}$ .

## 5) Memory.